UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/782,254	02/19/2004	Geary L. Eppley	MS1-1862US	4978

22801 7590 09/03/2009
LEE & HAYES, PLLC
601 W. RIVERSIDE AVENUE
SUITE 1400
SPOKANE, WA 99201

EXAMINER	
SYED, FARHAN M	

ART UNIT	PAPER NUMBER
2165	

NOTIFICATION DATE	DELIVERY MODE
09/03/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

lhptoms@leehayes.com

Office Action Summary	Application No.	Applicant(s)	
	10/782,254	EPPLEY ET AL.	
	Examiner	Art Unit	
	FARHAN M. SYED	2165	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 June 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,4 and 6-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4 and 6-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 and 3-33, filed 16 June 2009, are pending. The Examiner acknowledges amended claims 1, 6, 8, 12, and 15-23, and cancelled claims 3 and 5.

Response to Remarks/Argument

2. Applicant's arguments with respect to claims 1 and 3-33 have been considered but are moot in view of the new ground(s) of rejection.

The Examiner's rejections of the claims, now set forth are in light of the applicant's arguments against the art applied, But applied in the modified position therefore, the arguments are deemed moot.

Allowable Subject Matter

3. Claims 4, 10, 17, 18, and 25-26 objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 8-14 and 23-32 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As per claim 8, according to Applicant's disclosure, see page 1, lines 23-25, and at least page 7, lines 3-4, the filter engine appear to be software driven and therefore appear to lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se*.

Descriptive material can be characterized as either “functional descriptive material” or “nonfunctional descriptive material.” Both types of “descriptive material” are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)

Merely claiming nonfunctional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because “[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.”).

Claims 9-14 are dependent claims of independent claim 8 and do not appear to cure the deficiencies in claim 8 and therefore for reasons similar are rejected.

As per claim 23, according to Applicant's disclosure, see page 1, lines 23-25, and at least page 7, lines 3-4, the message processing system appear to be software driven and therefore appear to lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se*.

Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759. When <u>functional</u> descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)

Merely claiming <u>non</u>functional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because "[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.").

Claims 24-32 are dependent claims of independent claim 8 and do not appear to cure the deficiencies in claim 8 and therefore for reasons similar are rejected.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1, 6-9, 11-17, 19-24, and 27-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Campailla (U.S. 7,136,899, filed 11 December 2000)(previously presented) in view of a non-patent literature titled "Efficient Filtering of XML Documents for Selective Dissemination of Information," by Mehmet Altinel et al., 26th VLDB Conference, 2000, pages 53-64 (previously presented and known hereinafter as Altinel) and in view of a non-patent literature titled "On Efficient Matching of Streaming XML Documents and Queries" by Lakshmanan et al, University of British Columbia, Canada, 2002, pages 1-20 (previously presented and known hereinafter as Lakshmanan).

As per claim 1, Campailla teaches a method, comprising: receiving an input of data (i.e. *"The input message queue module receives the sequence of information messages from a publisher message generation system"* The preceding text clearly indicates that receiving of conforming input data is performed by receiving sequence of information messages from a publisher message generation system.) (see Figures 1 and 3; column 5, lines 50-57) that conforms to a query language(i.e. *"The broker server includes an input message queue module and a plurality of inverse query subscription modules."* The preceding text clearly indicates that the use of an inverse query module suggests using some form of query language.)(see Figures 1 and 3; column 5, lines 50-57) used by a filter engine (i.e. "inverse query modules" According to Applicant's disclosure, see page 1, lines 13-17, where filter engines may be called inverse query engines. The Examiner equates inverse query engines as inverse query modules.) (see Figures 1 and 3; column 5, lines 50-57) comprising two or more filter sub-engines (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n.)(see Figures 1 and 3), wherein at least one filter sub-engine is a general filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_1 is a general filter sub-engine.)(see Figures 1 and 3) and at least one filter sub-engine is an optimized filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_2 is an optimized filter sub-engine.)(see Figures 1 and 3).

Campailla does not explicitly teach wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine.

Altinel teaches wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine (i.e. *"For Xfilter, we implemented callback functions for parsing events of encounter: 1) a begin element tag; 2) an end element tag; or 3) data internal to an element. All of the handlers are passed the name and document level of the element for (or in) which the parsing event occurred."* The Examiner interprets that the XFilter provides a mechanism that determines whether the input conforms to grammar associated with the optimized filter sub-engine (i.e. parsed query resulting in optimized sub-filter engine.).)(Page 57, section 4.2, paragraph 3);

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel to include wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

Although the combination of Campailla and Altinel teaches the general aspect of the claimed limitation, it does not explicitly teach determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine;

determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language.

Lakshmanan teaches determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); directing the input to the optimized filter sub-engine for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby* *multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); and directing the input data to the second optimized filter sub-engine for processing (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4); if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel and with the further teachings of Lakshmanan to include determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

As per claim 2, Campailla teaches, wherein: the optimized filter sub-engine and the general sub-engine are encompassed by a single filter engine (i.e. *"Broker system encompasses the inverse query modules"*)(See Figure 3).

As per claim 6, Campailla teaches a method, further comprising: parsing the input to identify first and second sub-expressions (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); determining whether the first sub-expression can be processed by the optimized filter sub-engine (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); if the first sub-expression can be processed by the selective sub-engine, directing the first sub-expression to the optimized filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); if the first sub-expression cannot be processed by the optimized filter sub-engine, directing the first sub-expression to the general filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); determining whether the second sub-expression can be processed by the optimized filter sub-engine; if the second sub-expression can be processed by the optimized filter sub-engine (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45), directing the second sub-expression to the optimized filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and if the second sub-expression cannot be processed by the optimized filter sub-engine, directing the second sub-expression to the general filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 7, Campailla teaches a method, further comprising: obtaining a result of the processing of the first sub-expression (i.e. "*All the filters at a location step must evaluate to TRUE in order for the evaluation to continue to the descendant location steps.*") (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and processing the second sub-expression only if the result of the first sub-expression is true (i.e. "*All the filters at a location step must evaluate to TRUE in order for the evaluation to continue to the descendant location steps.*") (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 8 Campailla teaches a filter engine, comprising: receiving an input of data (i.e. *"The input message queue module receives the sequence of information messages from a publisher message generation system"* The preceding text clearly indicates that receiving of conforming input data is performed by receiving sequence of information messages from a publisher message generation system.) (see Figures 1 and 3; column 5, lines 50-57) that conforms to a query language(i.e. *"The broker server includes an input message queue module and a plurality of inverse query subscription modules."* The preceding text clearly indicates that the use of an inverse query module suggests using some form of query language.)(see Figures 1 and 3; column 5, lines 50-57) used by a filter engine (i.e. "inverse query modules" According to Applicant's disclosure, see page 1, lines 13-17, where filter engines may be called inverse query engines. The Examiner equates inverse query engines as inverse query modules.) (see Figures 1 and 3; column 5, lines 50-57) comprising two or more filter sub-engines (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n.)(see Figures 1 and 3), wherein at least one filter sub-engine is a general filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_1 is a general filter sub-engine.)(see Figures 1 and 3) and at least one filter sub-engine is an optimized filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_2 is an optimized filter sub-engine.)(see Figures 1 and 3).

Campailla does not explicitly teach wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine.

Altinel teaches wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine (i.e. *"For Xfilter, we implemented callback functions for parsing events* *of encounter: 1) a begin element tag; 2) an end element tag; or 3) data internal to an element. All of the handlers are passed the name and document level of the element for (or in) which the parsing event occurred."* The Examiner interprets that the XFilter provides a mechanism that determines whether the input conforms to grammar associated with the optimized filter sub-engine (i.e. parsed query resulting in optimized sub-filter engine.).)(Page 57, section 4.2, paragraph 3);

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel to include wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

Although the combination of Campailla and Altinel teaches the general aspect of the claimed limitation, it does not explicitly teach determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language.

Lakshmanan teaches determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); directing the input to the optimized filter sub-engine for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4;

Figure 1); and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); and directing the input data to the second optimized filter sub-engine for processing (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4); if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel and with the further teachings of Lakshmanan to include determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

As per claim 9, Campailla teaches a filter engine, wherein the analyzer is further configured to analyze a new filter added to the filter engine and to determine an appropriate matcher with which to associate the new filter (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 11, the combination of Campailla, Graefe, and Lakshmanan do not explicitly teach wherein the determining further comprises recognizing whether or not the input conforms to a grammar of the optimized filter sub-engine.

Altinel teaches wherein the determining further comprises recognizing whether or not the input conforms to a grammar of the optimized filter sub-engine (i.e. *"For Xfilter, we implemented callback functions for parsing events of encounter: 1) a begin element tag; 2) an end element tag; or 3) data internal to an element. All of the handlers are passed the name and document level of the element for (or in) which the parsing event occurred."*)(Page 57, section 4.2, paragraph 3).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla, Graefe, and Lashmanan with the teachings of Altinel to include wherein the determining further comprises recognizing whether or not the input conforms to a grammar of the optimized filter sub-engine with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications (Graefe, page 359).

As per claim 12, Campailla teaches a method, further comprising: parsing the input to identify first and second sub-expressions (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); determining whether the first sub-expression can be processed by the optimized filter sub-engine (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15;

column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); if the first sub-expression can be processed by the selective sub-engine, directing the first sub-expression to the optimized filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); if the first sub-expression cannot be processed by the optimized filter sub-engine, directing the first sub-expression to the general filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); determining whether the second sub-expression can be processed by the optimized filter sub-engine; if the second sub-expression can be processed by the optimized filter sub-engine (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45), directing the second sub-expression to the optimized filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and if the second sub-expression cannot be processed by the optimized filter sub-engine, directing the second sub-expression to the general filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 13, Campailla teaches a method, further comprising: obtaining a result of the processing of the first sub-expression (i.e. "*All the filters at a location step must evaluate to TRUE in order for the evaluation to continue to the descendant location steps.*") (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and processing the second sub-expression only if the result of the first sub-expression is true (i.e. "*All the filters at a location step must evaluate to TRUE in order for the evaluation to continue to the descendant location steps.*") (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 14, Campailla teaches a filter engine, wherein the at least one optimized matcher further comprises: a first selective sub-engine configured to process inputs that conform to a first subset of the input language (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); a second selective sub-engine configured to process inputs that conform to a second subset of the input language (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and wherein the first subset and the second subset are unique subsets of the input language (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 15, Campailla teaches one or more computer-readable storage media containing computer-executable instructions that, when executed direct a computer system to: receiving an input of data (i.e. *"The input message queue module receives the sequence of information messages from a publisher message generation system"* The preceding text clearly indicates that receiving of conforming input data is performed by receiving sequence of information messages from a publisher message generation system.) (see Figures 1 and 3; column 5, lines 50-57) that conforms to a query language(i.e. *"The broker server includes an input message queue module and a plurality of inverse query subscription modules."* The preceding text clearly indicates that the use of an inverse query module suggests using some form of query language.)(see Figures 1 and 3; column 5, lines 50-57) used by a filter engine (i.e. "inverse query modules" According to Applicant's disclosure, see page 1, lines 13-17, where filter engines may be called inverse query engines. The Examiner equates inverse query engines as inverse query modules.) (see Figures 1 and 3; column 5, lines 50-57) comprising two or more filter sub-engines (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n.)(see Figures 1 and 3), wherein at least one filter sub-engine is a general filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_1 is a general filter sub-engine.)(see Figures 1 and 3) and at least one filter sub-engine is an optimized filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_2 is an optimized filter sub-engine.)(see Figures 1 and 3).

Campailla does not explicitly teach wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine.

Altinel teaches wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine (i.e. *"For Xfilter, we implemented callback functions for parsing events of encounter: 1) a begin element tag; 2) an end element tag; or 3) data internal to an element. All of the handlers are passed the name and document level of the element for (or in) which the parsing event occurred."* The Examiner interprets that the XFilter provides a mechanism that determines whether the input conforms to grammar associated with the optimized filter sub-engine (i.e. parsed query resulting in optimized sub-filter engine.).)(Page 57, section 4.2, paragraph 3);

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel to include wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

Although the combination of Campailla and Altinel teaches the general aspect of the claimed limitation, it does not explicitly teach determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language.

Lakshmanan teaches determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4;

Figure 1); directing the input to the optimized filter sub-engine for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); and directing the input data to the second optimized filter sub-engine for processing (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one* *at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4); if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel and with the further teachings of Lakshmanan to include determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

As per claim 16, Campailla teaches a computer-readable media, further comprising the step of accepting input messages for both the selective sub-engine and the general sub-engine by way of a single input means so that an input message sending application does not have to distinguish between the selective sub-engine and the general sub-engine (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 19, Campailla teaches a method, further comprising: parsing the input to identify first and second sub-expressions (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); determining whether the first sub-expression can be processed by the optimized filter sub-engine (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); if the first sub-expression can be processed by the selective sub-engine, directing the first sub-expression to the optimized filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); if the first sub-expression cannot be processed by the optimized filter sub-engine, directing the first sub-expression to the general filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column

7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); determining whether the second sub-expression can be processed by the optimized filter sub-engine; if the second sub-expression can be processed by the optimized filter sub-engine (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45), directing the second sub-expression to the optimized filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and if the second sub-expression cannot be processed by the optimized filter sub-engine, directing the second sub-expression to the general filter sub-engine for processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 20, Campailla teaches a computer-readable storage media, further comprising the step of deriving a final result of the input message processing from at least one result of the sub-expression processing (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 21, Campailla teaches a method, further comprising: obtaining a result of the processing of the first sub-expression (i.e. "*All the filters at a location step must evaluate to TRUE in order for the evaluation to continue to the descendant location steps.*") (See Figures

1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and processing the second sub-expression only if the result of the first sub-expression is true (i.e. "*All the filters at a location step must evaluate to TRUE in order for the evaluation to continue to the descendant location steps.*") (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 23, Campailla teaches a message processing system, comprising: receiving an input of data (i.e. *"The input message queue module receives the sequence of information messages from a publisher message generation system"* The preceding text clearly indicates that receiving of conforming input data is performed by receiving sequence of information messages from a publisher message generation system.) (see Figures 1 and 3; column 5, lines 50-57) that conforms to a query language(i.e. *"The broker server includes an input message queue module and a plurality of inverse query subscription modules."* The preceding text clearly indicates that the use of an inverse query module suggests using some form of query language.)(see Figures 1 and 3; column 5, lines 50-57) used by a filter engine (i.e. "inverse query modules" According to Applicant's disclosure, see page 1, lines 13-17, where filter engines may be called inverse query engines. The Examiner equates inverse query engines as inverse query modules.) (see Figures 1 and 3; column 5, lines 50-57) comprising two or more filter sub-engines (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n.)(see Figures 1 and 3), wherein at least one filter sub-engine is a general filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_1 is a general filter sub-engine.)(see Figures 1 and 3) and at least one filter sub-engine is an optimized filter sub-engine (see Figure 3 that illustrates two or more filter sub-engines, i.e. inverse query sub_1, inverse query sub_2, …inverse query sub_n, and where inverse query sub_2 is an optimized filter sub-engine.)(see Figures 1 and 3).

Campailla does not explicitly teach wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine.

Altinel teaches wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine (i.e. *"For Xfilter, we implemented callback functions for parsing events of encounter: 1) a begin element tag; 2) an end element tag; or 3) data internal to an element. All of the handlers are passed the name and document level of the element for (or in) which the parsing event occurred."* The Examiner interprets that the XFilter provides a mechanism that determines whether the input conforms to grammar associated with the optimized filter sub-engine (i.e. parsed query resulting in optimized sub-filter engine.).)(Page 57, section 4.2, paragraph 3);

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel to include wherein the determining whether the input data conforms to a grammar associated with the optimized filter sub-engine, wherein the optimized query filter sub-engine is configured to handle only a subset of the query language handled by the general filter sub-engine with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

Although the combination of Campailla and Altinel teaches the general aspect of the claimed limitation, it does not explicitly teach determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language.

Lakshmanan teaches determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); directing the input to the optimized filter sub-engine for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); and directing the input data to the second optimized filter sub-engine for processing (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4); if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing (i.e. *"A more clever approach is to devise algorithms that make a constent number of passes over the document and determine the queries answered by each of its elements. This will permit set-oriented processing whereby multiple queries are processed together. Such an algorithm is non-trivial since: (i) queries mayhave repeating tags and (ii) the same query may have multiple matchings into a given document. Both these features are illustrated in Figure 1."* The preceding text clearly suggests that a selective sub-engine occurs in the background that produces multiple matchings in a given document.)(Lakshmanan, page 4; Figure 1); if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language (i.e. "We have implemented a MatchMaker system for matching XML documents to queries and for providing notification service. As an overview, XML data streams through the MatchMaker, with which users have registered their requirements in the form of queries, in a requirements registry. The MatchMaker consults the registry in determining which users a given data element is relevant to." *"A naïve way to obtain these labels is to process the user queries, one at a time, finding all its matchings, and compile the answers into appropriate label sets for the document nodes. This strategy is very inefficient as it makes a number of passes over the given document, proportional to the number of queries."* The preceding text clearly indicates that a general sub-engine is a user queries that is used to find all matchings. Unlike a specific sub-engine that returns selected matchings, a general sub-engine, akin to a user queries performs a general search that retrieves all matchings.)(Lashmanan, pages 3-4).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Campailla with the teachings of Altinel and with the further teachings of Lakshmanan to include determining whether the input data can be processed by the optimized first sub-engine or by the second sub-engine, the determining comprising identifying if the input data comprises a subset of the query language; directing the input to the optimized filter sub-engine for processing; and if the determining indicates that the input cannot be processed by the first sub-engine; determining whether the input data can be processed by a second optimized filter sub-engine, wherein the second optimized filter sub-engine is configured to handle only a subset of the query language, and wherein the subset of the query language that the second optimized filter sub-engine is configured to handle excludes the subset of the query language that the first optimized filter sub-engine is configured to handle; and directing the input data to the second optimized filter sub-engine for processing; if the determining indicates that the input can be processed by the optimized filter sub-engine, then directing the input to the optimized filter sub-engine for optimized filter for processing; if the determining indicates that the input cannot be processed by the optimized filter sub-engine, then directing the input to the general sub-engine for processing, wherein the general filter sub-engine is configured to handle all aspects of the language with the motivation to provide more flexibility and better performance for both conventional and non-conventional database management systems and applications.

As per claim 22, Campailla teaches a computer-readable storage media, wherein each matcher includes a set of queries against which input messages directed to the respective matchers are tried, and wherein each set of queries is unique (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 24, Campailla teaches a message processing system, wherein: the optimized filter processor further comprises a first set of queries against which a message directed to the optimized filter processor is compared (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); the general filter processor further comprises a second set of queries against which a message directed to the general filter processor is compared; and the first set of queries contains fewer queries than the second set of queries (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and

33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 27, Campailla teaches a message processing system, wherein the optimized filter processor further comprises means for optimizing message processing over the set of queries included in the optimized filter processor (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 28, Campailla teaches a message processing system, wherein the means for optimizing message processing further comprises a hash function (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 29, Campailla teaches a message processing system, wherein: the optimized filter processor is a first filter processor; and the message processing system further comprises a second optimized filter processor to which messages may be directed, the second optimized filter processor supporting a unique subset of the query language (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45); and the distribution means is further configured to direct the message to the second optimized filter processor if the first optimized filter processor cannot process the message but the second optimized filter processor can process the message (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 30, Campailla teaches a message processing system, further comprising means for parsing the message into constituent sub-expressions, and the analyzing means is further configured to process individual sub-expression as an individual message and to evaluate sub-expression processing results to derive a result corresponding to the message (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 31, Campailla teaches a message processing system, wherein the message is a sub-expression of a parent message (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 32, Campailla teaches a message processing system, further comprising means for determining whether a filter in the system is associated with the generalized filter processor or the optimized filter processor (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

As per claim 33, Campailla teaches determining comprises generating a hash of the input data in order to determine if an optimized sub-engine is capable of handling the input data (See Figures 1, 3, 4, 6, and 8; column 4, lines 5-15 and 33-67; column 5, lines 50-67; column 6, lines 1-67; column 7, lines 1-15; column 7, lines 43-56; column 9, lines 15-40 and 55-67; and column 10, lines 15-45).

Conclusion

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhan M. Syed whose telephone number is 571-272-7191. The examiner can normally be reached on 8:30AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Neveen Abel Jalil can be reached on 571-272-4074. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/F. M. S./
Examiner, Art Unit 2165

/Neveen Abel-Jalil/

Supervisory Patent Examiner, Art Unit 2165